

CT484: PHÁT TRIỂN ỨNG DỤNG DI ĐỘNG

XÂY DỰNG ỨNG DỤNG MYSHOP - PHẦN 2

File báo cáo cần nộp là file PDF trong đó có ghi thông tin **mã sinh viên, họ tên, lớp học** phần cùng với **hình minh họa tại các bước kiểm tra kết quả thực thi, các chức năng (không cần chụp hình mã nguồn)**. Cuối file báo cáo ghi đường link đến GitHub mã nguồn của dự án.

Ứng dụng MyShop có các chức năng chính sau:

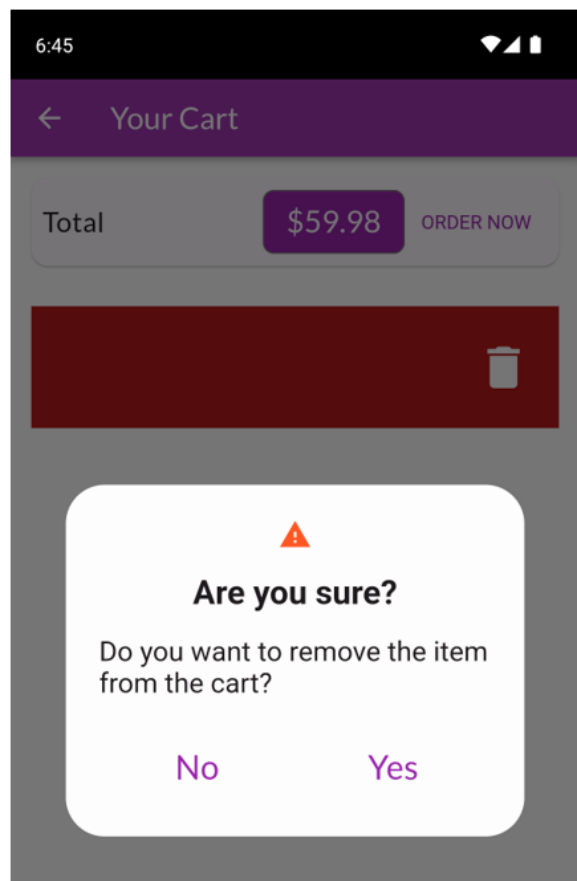
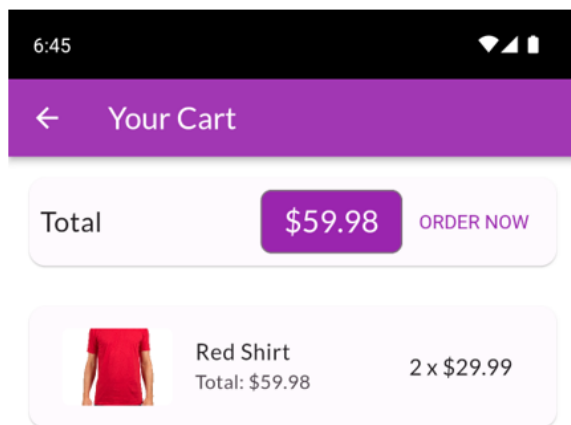
- Hiển thị và cập nhật danh mục các sản phẩm
- Xem chi tiết một sản phẩm
- Đánh dấu sản phẩm được yêu thích
- Thêm, xóa sản phẩm trong giỏ hàng
- Thực hiện đặt hàng, xem lại đơn hàng, chi tiết đơn hàng
- Đăng ký, đăng nhập
- Lưu trữ dữ liệu trên Firebase

Sinh viên chỉ cần tạo MỘT báo cáo duy nhất cho tất cả các buổi thực hành.

(Tiếp tục từ kết quả phần 1)

File báo cáo buổi 2 làm tiếp tục từ file báo cáo buổi 1.

Bước 1: Xây dựng trang hiển thị giỏ hàng



- Định nghĩa lớp **CartItem** miêu tả thông tin một mặt hàng trong giỏ hàng (*lib/models/cart_item.dart*):

```
class CartItem {  
  final String id;  
  final String title;  
  final String imageUrl;  
  final int quantity;  
  final double price;  
  
  CartItem({  
    required this.id,  
    required this.title,  
    required this.imageUrl,  
    required this.quantity,  
    required this.price,  
  });  
  
  CartItem copyWith({  
    String? id,  
    String? title,  
    String? imageUrl,  
    int? quantity,  
    double? price,  
  }) {  
    return CartItem(  
      id: id ?? this.id,  
      title: title ?? this.title,  
      imageUrl: imageUrl ?? this.imageUrl,  
      quantity: quantity ?? this.quantity,  
      price: price ?? this.price,  
    );  
  }  
}
```

- Định nghĩa lớp **CartManager** quản lý các mặt hàng trong giỏ hàng (*lib/ui/cart/cart_manager.dart*):

```
import '../models/cart_item.dart';

class CartManager {
  final Map<String, CartItem> _items = {
    'p1': CartItem(
      id: 'c1',
      title: 'Red Shirt',
      imageUrl:
        'https://cdn.pixabay.com/photo/2016/10/02/22/17/red-t-shirt-1710578_1280.jpg',
      price: 29.99,
      quantity: 2,
    ),
  };

  int get productCount {
    return _items.length;
  }

  List<CartItem> get products {
    return _items.values.toList();
  }

  Iterable<MapEntry<String, CartItem>> get productEntries {
    return {..._items}.entries;
  }

  double get totalAmount {
    var total = 0.0;
    _items.forEach((key, cartItem) {
      total += cartItem.price * cartItem.quantity;
    });
    return total;
  }
}
```

- Định nghĩa thư viện các hàm tiện ích về hộp thoại (*lib/ui/shared/dialog_utils.dart*):

```

import 'package:flutter/material.dart';

Future<bool?> showConfirmDialog(BuildContext context, String message) {
  return showDialog(
    context: context,
    builder: (ctx) => AlertDialog(
      icon: const Icon(Icons.warning),
      title: const Text('Are you sure?'),
      content: Text(message),
      actions: <Widget>[
        Row(
          mainAxisAlignment: MainAxisAlignment.spaceEvenly,
          children: [
            Expanded(
              child: ActionButton(
                actionText: 'No',
                onPressed: () {
                  Navigator.of(ctx).pop(false);
                },
              ),
            Expanded(
              child: ActionButton(
                actionText: 'Yes',
                onPressed: () {
                  Navigator.of(ctx).pop(true);
                },
              ),
            ),
          ],
        ),
      ],
    ),
  );
}

```

Trong đó, widget **ActionButton** được định nghĩa như sau (vẫn trong *lib/ui/shared/dialog_utils.dart*):

```

class ActionButton extends StatelessWidget {
  const ActionButton({
    super.key,
    this.actionText,
    this.onPressed,
  });

  final String? actionText;
  final void Function()? onPressed;

  @override
  Widget build(BuildContext context) {
    return TextButton(
      onPressed: onPressed,
      child: Text(

```

```

        actionText ?? 'Okay',
        style: Theme.of(context).textTheme.titleMedium!.copyWith(
          color: Theme.of(context).colorScheme.primary,
          fontSize: 24,
        ),
      ),
    );
  }
}

```

Hàm [showDialog](#)(context, builder): hiển thị một hộp thoại phía trên nội dung hiện thời. Tham số builder là hàm tạo widget Dialog. Hai dạng widget Dialog phổ biến: [AlertDialog](#) và [SimpleDialog](#). Các thuộc tính quan trọng của một **AlertDialog** là **title**: tựa đề hội thoại, **content**: nội dung chính hội thoại và **actions**: dòng các nút tương tác với hội thoại.

Hiệu chỉnh **ThemeData** của **MaterialApp** trong *lib/main.dart* để tùy chỉnh theme cho dialog:

```

...
class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    ...
    final themeData = ThemeData(
      ...
      // Thêm định nghĩa dialogTheme vào ThemeData
      dialogTheme: DialogTheme(
        titleTextStyle: TextStyle(
          color: colorScheme.onBackground,
          fontSize: 24,
          fontWeight: FontWeight.bold,
        ),
        contentTextStyle: TextStyle(
          color: colorScheme.onBackground,
          fontSize: 20,
        ),
      ),
    ),
  ),
),

  return MaterialApp(
    title: 'My Shop',
    debugShowCheckedModeBanner: false,
    theme: themeData,
    ...
  );
}
}

```

- Định nghĩa widget **CartItemCard** hiển thị thông tin một mặt hàng trong giỏ hàng (*lib/ui/cart/cart_item_card.dart*):

```

import 'package:flutter/material.dart';

import '../models/cart_item.dart';
import '../shared/dialog_utils.dart';

class CartItemCard extends StatelessWidget {
  final String productId;
  final CartItem cartItem;

  const CartItemCard({
    required this.productId,
    required this.cartItem,
    super.key,
  });

  @override
  Widget build(BuildContext context) {
    return Dismissible(
      key: ValueKey(cartItem.id),
      background: Container(
        color: Theme.of(context).colorScheme.error,
        alignment: Alignment.centerRight,
        padding: const EdgeInsets.only(right: 20),
        margin: const EdgeInsets.symmetric(
          horizontal: 15,
          vertical: 4,
        ),
      ),
      child: const Icon(
        Icons.delete,
        color: Colors.white,
        size: 40,
      ),
    ),
    direction: DismissDirection.endToStart,
    confirmDismiss: (direction) {
      return showConfirmDialog(
        context,
        'Do you want to remove the item from the cart?',
      );
    },
    onDismissed: (direction) {
      print('Cart item dismissed');
    },
    child: ItemInfoCard(),
  );
}

```

```

}

class ItemInfoCard extends StatelessWidget {
  const ItemInfoCard({super.key});

  @override
  Widget build(BuildContext context) {

  }
}

```

Widget [Dismissible](#) là widget có thể được loại bỏ (dismiss) bằng cách vuốt theo hướng chỉ định. Một số thuộc tính quan trọng của Dismissable: **key** thuộc tính dùng định danh widget; **direction** chỉ định hướng vuốt widget (DismissDirection); **background** widget nền cho widget con; **confirmDismiss** hàm được gọi để xác nhận loại bỏ, trả về true widget sẽ được loại bỏ, ngược lại quay về vị trí cũ; **onDismissed** hàm được gọi sau khi widget đã được loại bỏ.

Widget *ItemInfoCard* được định nghĩa như sau:

```

class ItemInfoCard extends StatelessWidget {
  const ItemInfoCard(
    this.cartItem, {
    super.key,
  });

  final CartItem cartItem;

  @override
  Widget build(BuildContext context) {
    return Card(
      margin: const EdgeInsets.symmetric(
        horizontal: 15,
        vertical: 4,
      ),
      child: Padding(
        padding: const EdgeInsets.all(8),
        child: ListTile(
          leading: ClipRRect(
            borderRadius: BorderRadius.circular(5),
            child: Image.network(
              cartItem.imageUrl,
              fit: BoxFit.cover,
              width: 80,
              height: 80,
            ),
          ),
          title: Text(cartItem.title),
          subtitle: Text('Total: \${(cartItem.price * cartItem.quantity)}'),
          trailing: Text(
            '${cartItem.quantity} x \${cartItem.price}',
            style: Theme.of(context).textTheme.titleMedium,
          ),
        ),
      ),
    );
  }
}

```

Widget [FittedBox](#) co giãn và bố trí con của nó theo tiêu chí được chỉ định bởi thuộc tính fit (mặc định là BoxFit.contain). Hiệu chỉnh **CartItemCard**:

```

class CartItemCard extends StatelessWidget {
  ...

  @override
  Widget build(BuildContext context) {
    return Dismissible(
      ...
      // Cập nhật ItemInfoCard
      child: ItemInfoCard(cartItem),
    );
  }
}

```

- Định nghĩa trang hiển thị thông tin một giỏ hàng (*lib/ui/cart/cart_screen.dart*):


```

import 'package:flutter/material.dart';

import 'cart_manager.dart';
import 'cart_item_card.dart';

class CartScreen extends StatelessWidget {
  const CartScreen({super.key});

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Your Cart'),
      ),
      body: Column(
        children: <Widget>[
          CartSummary(),
          const SizedBox(height: 10),
          Expanded(
            child: CartItemList(),
          )
        ],
      ),
    );
  }
}

class CartItemList extends StatelessWidget {
  const CartItemList({super.key});

  @override
  Widget build(BuildContext context) {
  }
}

```

```

class CartSummary extends StatelessWidget {
  const CartSummary({super.key});

  @override
  Widget build(BuildContext context) {

  }
}

```

Widget *CartItem* và *CartSummary* được định nghĩa như sau:

```

class CartItemList extends StatelessWidget {
  const CartItemList(
    this.cart, {
    super.key,
  });

  final CartManager cart;

  @override
  Widget build(BuildContext context) {
    return ListView(
      children: cart.productEntries
        .map(
          (entry) => CartItemCard(
            productId: entry.key,
            cartItem: entry.value,
          ),
        )
        .toList(),
    );
  }
}

```

```

class CartSummary extends StatelessWidget {
  const CartSummary({
    super.key,
    required this.cart,
    this.onOrderNowPressed,
  });

  final CartManager cart;
  final void Function()? onOrderNowPressed;

  @override
  Widget build(BuildContext context) {
    return Card(
      margin: const EdgeInsets.all(15),
      child: Padding(
        padding: const EdgeInsets.all(8),
        child: Row(
          mainAxisAlignment: MainAxisAlignment.spaceBetween,
          children: <Widget>[
            const Text(
              'Total',
              style: TextStyle(fontSize: 20),
            ),
            const Spacer(),
            Chip(
              label: Text(
                '\${cart.totalAmount.toStringAsFixed(2)}',
                style: Theme.of(context).primaryTextTheme.titleLarge,
              ),
              backgroundColor: Theme.of(context).colorScheme.primary,
            ),
            TextButton(
              onPressed: onOrderNowPressed,
              style: TextButton.styleFrom(
                textStyle: TextStyle(
                  color: Theme.of(context).colorScheme.primary,
                ),
              ),
              child: const Text('ORDER NOW'),
            ),
          ],
        ),
      ),
    );
  }
}

```

Hiệu chỉnh **CartScreen**:

```

class CartScreen extends StatelessWidget {
  ...

  @override

```

```

Widget build(BuildContext context) {
  // Tạo đối tượng CartManager
  final cart = CartManager();

  return Scaffold(
    ...
    body: Column(
      children: <Widget>[
        // Cập nhật CartSummary
        CartSummary(
          cart: cart,
          onOrderNowPressed: () {
            print('An order has been added');
          },
        ),
        const SizedBox(height: 10),
        Expanded(
          // Cập nhật CartItemList
          child: CartItemList(cart),
        ),
      ],
    ),
  );
}

```

- Hiệu chỉnh *lib/main.dart* kiểm tra trang hiển thị giỏ hàng:

```

...
import 'ui/cart/cart_screen.dart';
...
class MyApp extends StatelessWidget {
  ...
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      ...
      // Hiệu chỉnh trang home
      home: const SafeArea(
        child: CartScreen(),
      ),
    );
  }
}

```

- Sau khi kiểm tra, lưu mã nguồn vào repo git và lên GitHub:

```

git add -u
git add lib/models lib/ui
git commit -m "Xây dựng trang giỏ hàng"
git push origin master

```

Bước 2: Xây dựng trang hiển thị các đặt hàng



- Định nghĩa lớp **OrderItem** miêu tả thông tin một đặt hàng (*lib/models/order_item.dart*):

```

1  import 'cart_item.dart';
2
3  class OrderItem {
4    final String? id;
5    final double amount;
6    final List<CartItem> products;
7    final DateTime dateTime;
8
9    int get productCount {
10     return products.length;
11   }
12
13   OrderItem({
14     this.id,
15     required this.amount,
16     required this.products,
17     DateTime? dateTime,
18   }) : dateTime = dateTime ?? DateTime.now();
19
20   OrderItem copyWith({
21     String? id,
22     double? amount,
23     List<CartItem>? products,
24     DateTime? dateTime,
25   }) {
26     return OrderItem(
27       id: id ?? this.id,
28       amount: amount ?? this.amount,
29       products: products ?? this.products,
30       dateTime: dateTime ?? this.dateTime,
31     );
32   }
33 }

```

- Định nghĩa lớp **OrderManager** quản lý các đặt hàng (*lib/ui/orders/order_manager.dart*):

```

import '../models/cart_item.dart';
import '../models/order_item.dart';

class OrdersManager {
  final List<OrderItem> _orders = [
    OrderItem(
      id: 'o1',
      amount: 59.98,
      products: [
        CartItem(
          id: 'c1',
          title: 'Red Shirt',
          imageUrl:
            'https://cdn.pixabay.com/photo/2016/10/02/22/17/red-t-shirt-1710578_1280.jpg',
          price: 29.99,
          quantity: 2,
        )
      ],
      dateTime: DateTime.now(),
    )
  ];

  int get orderCount {
    return _orders.length;
  }

  List<OrderItem> get orders {
    return [..._orders];
  }
}

```

- Định nghĩa widget **OrderItemCard** hiển thị thông tin một đặt hàng (*lib/ui/orders/order_item_card.dart*):

```

import 'dart:math';

import 'package:flutter/material.dart';
import 'package:intl/intl.dart';

import '../models/order_item.dart';

class OrderItemCard extends StatefulWidget {
  final OrderItem order;

  const OrderItemCard(this.order, {super.key});

  @override
  State<OrderItemCard> createState() => _OrderItemCardState();
}

class _OrderItemCardState extends State<OrderItemCard> {
  var _expanded = false;

  @override
  Widget build(BuildContext context) {
    return Card(
      margin: const EdgeInsets.all(10),
      child: Column(
        children: <Widget>[
          OrderSummary(),
          if (_expanded) OrderItemList()
        ],
      ),
    );
  }
}

class OrderItemList extends StatelessWidget {
  const OrderItemList({super.key});

  @override
  Widget build(BuildContext context) {

  }
}

class OrderSummary extends StatelessWidget {
  const OrderSummary({super.key});

  @override
  Widget build(BuildContext context) {

  }
}

```



```
}
```

Hàm *OrderItemList* và *OrderSummary* được định nghĩa như sau:

```
class OrderItemList extends StatelessWidget {
  const OrderItemList(
    this.order, {
    super.key,
  });

  final OrderItem order;

  @override
  Widget build(BuildContext context) {
    return Container(
      padding: const EdgeInsets.symmetric(horizontal: 15, vertical: 4),
      height: min(order.productCount * 20.0 + 10, 100),
      child: ListView(
        children: order.products
          .map(
            (prod) => Row(
              mainAxisAlignment: MainAxisAlignment.spaceBetween,
              children: <Widget>[
                Text(
                  prod.title,
                  style: const TextStyle(
                    fontSize: 18,
                    fontWeight: FontWeight.bold,
                  ),
                ),
                Text(
                  '${prod.quantity}x ${prod.price}',
                  style: const TextStyle(
                    fontSize: 18,
                    color: Colors.grey,
                  ),
                ),
              ],
            ),
          ).toList(),
      ),
    );
  }
}
```

```

class OrderSummary extends StatelessWidget {
  const OrderSummary({
    super.key,
    required this.order,
    required this.expanded,
    this.onExpandPressed,
  });

  final bool expanded;
  final OrderItem order;
  final void Function()? onExpandPressed;

  @override
  Widget build(BuildContext context) {
    return ListTile(
      titleTextStyle: Theme.of(context).textTheme.titleLarge,
      title: Text('\${order.amount.toStringAsFixed(2)}'),
      subtitle: Text(
        DateFormat('dd/MM/yyyy hh:mm').format(order.dateTime),
      ),
      trailing: IconButton(
        icon: Icon(expanded ? Icons.expand_less : Icons.expand_more),
        onPressed: onExpandPressed,
      ),
    );
  }
}

```

Widget *OrderSummary* trong **OrderItemCard** sử dụng thư viện `intl` để định dạng ngày tháng, khai báo thư viện trong *pubspec.yaml* (chọn phiên bản mới nhất):

```

...
dependencies:
  flutter:
    sdk: flutter
  intl: ^0.18.1
...

```

(Hoặc có thể thêm thư viện `intl` bằng cách thực thi câu lệnh sau: `flutter pub add intl`).

Hiệu chỉnh **OrderItemCard**:

```

class _OrderItemCardState extends State<OrderItemCard> {
  var _expanded = false;

  @override
  Widget build(BuildContext context) {
    return Card(
      margin: const EdgeInsets.all(10),
      child: Column(
        children: <Widget>[
          // Hiệu chỉnh OrderSummary
          OrderSummary(

```

```

        expanded: _expanded,
        order: widget.order,
        onExpandPressed: () {
          setState(() {
            _expanded = !_expanded;
          });
        },
      ),
      // Hiệu chỉnh OrderItemList
      if (_expanded) OrderItemList(widget.order)
    ],
  ),
);
}
}

```

- Định nghĩa trang hiển thị thông tin các đặt hàng (*lib/ui/orders/orders_screen.dart*):

```

import 'package:flutter/material.dart';

import 'orders_manager.dart';
import 'order_item_card.dart';

class OrdersScreen extends StatelessWidget {
  const OrdersScreen({super.key});

  @override
  Widget build(BuildContext context) {
    final ordersManager = OrdersManager();

    return Scaffold(
      appBar: AppBar(
        title: const Text('Your Orders'),
      ),
      body: ListView.builder(
        itemCount: ordersManager.orderCount,
        itemBuilder: (ctx, i) => OrderItemCard(ordersManager.orders[i]),
      ),
    );
  }
}

```

- Hiệu chỉnh *lib/main.dart* kiểm tra trang hiển thị các đặt hàng:

```

...
import 'ui/orders/orders_screen.dart';
...
class MyApp extends StatelessWidget {
  ...
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      ...
      // Hiệu chỉnh trang home

```

```

        home: const SafeArea(
          child: ordersScreen(),
        ),
      );
    }
  }
}

```

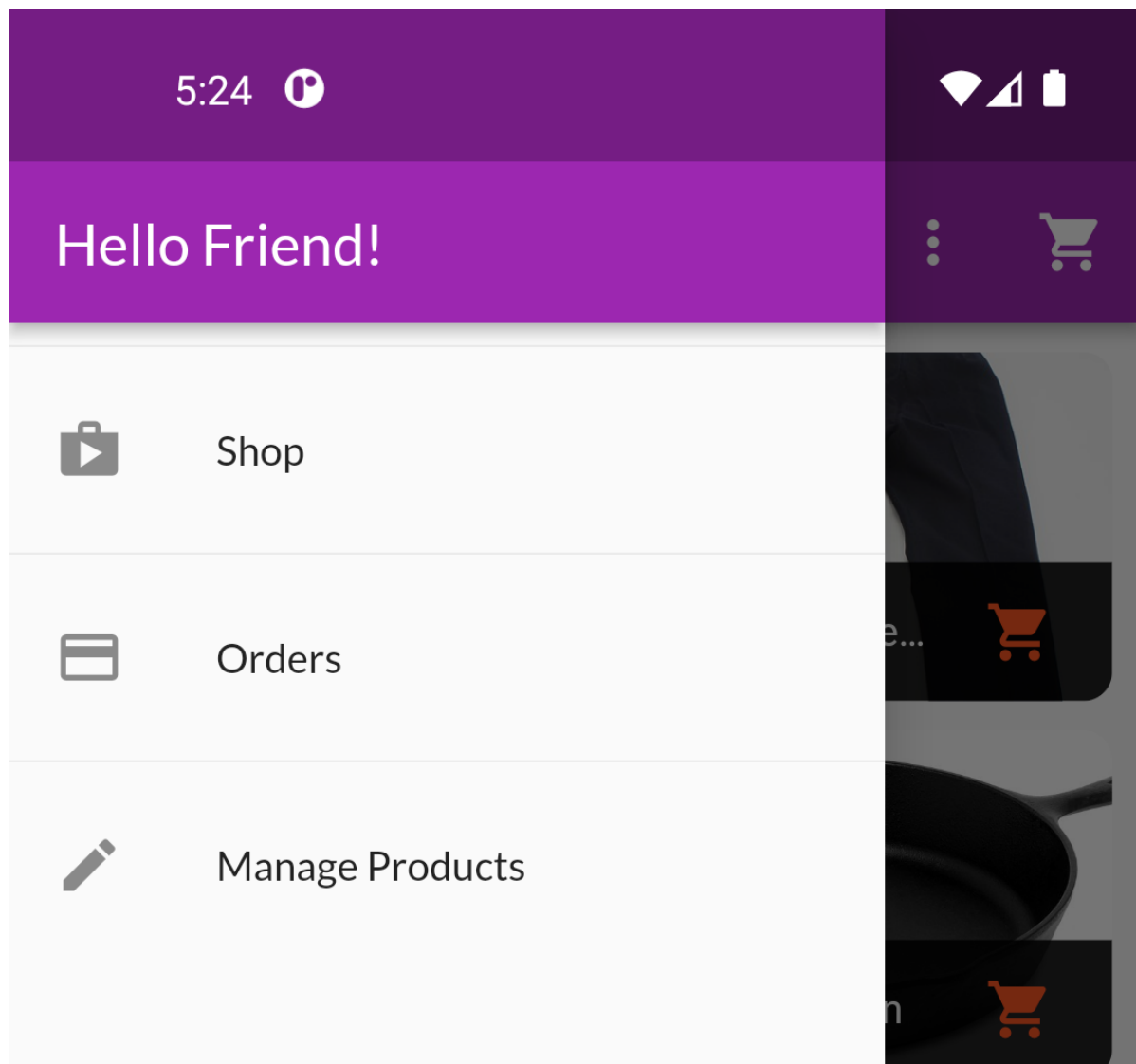
- Sau khi kiểm tra, lưu mã nguồn vào repo git và lên GitHub:

```

git add -u
git add lib/models lib/ui
git commit -m "xay dung trang cac dat hang"
git push origin master

```

Bước 3: Định tuyến theo tên



- Tạo tập tin `lib/ui/screens.dart` là tập tin import tập trung của thư mục `lib/ui`:

```

export 'products/products_overview_screen.dart';
export 'products/product_detail_screen.dart';
export 'products/user_products_screen.dart';
export 'products/products_manager.dart';

export 'orders/orders_screen.dart';
export 'orders/orders_manager.dart';

export 'cart/cart_screen.dart';
export 'cart/cart_manager.dart';

```

Hiệu chỉnh tập tin *lib/main.dart*, thay vì import từng tập tin *_screen riêng lẻ, chỉ cần import tập tin *lib/ui/screens.dart*

- Đặt tên cho các trang màn hình:
 - *lib/ui/products/product_detail_screen.dart*:

```

class ProductDetailScreen extends StatelessWidget {
  static const routeName = '/product-detail';
  ...
}

```

- *lib/ui/products/user_products_screen.dart*:

```

class UserProductsScreen extends StatelessWidget {
  static const routeName = '/user-products';
  ...
}

```

- *lib/ui/cart/cart_screen.dart*:

```

class CartScreen extends StatelessWidget {
  static const routeName = '/cart';
  ...
}

```

- *lib/ui/orders/orders_screen.dart*:

```

class OrdersScreen extends StatelessWidget {
  static const routeName = '/orders';
  ...
}

```

- Hiệu chỉnh *lib/main.dart*, khai báo các trang của ứng dụng:

```

class MyApp extends StatelessWidget {
  ...
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      ...
      home: const ProductsOverviewScreen(),
    );
  }
}

```

```
// Thuộc tính routes thường dùng dùng khai báo
// các route không tham số.
routes: {
  CartScreen.routeName: (ctx) => const SafeArea(
    child: CartScreen(),
  ),
  OrdersScreen.routeName: (ctx) => const SafeArea(
    child: OrdersScreen(),
  ),
  UserProductsScreen.routeName: (ctx) => const SafeArea(
    child: UserProductsScreen(),
  ),
},
// onGenerateRoute sẽ được gọi khi không tìm thấy route yêu cầu
// trong thuộc tính routes ở trên. Thường dùng để truyền tham số
// hoặc tùy biến hiệu ứng chuyển trang.
onGenerateRoute: (settings) {
  if (settings.name == ProductDetailScreen.routeName) {
    final productId = settings.arguments as String;
    return MaterialPageRoute(
      settings: settings,
      builder: (ctx) {
        return SafeArea(
          child: ProductDetailScreen(
            ProductsManager().findById(productId!),
          ),
        );
      },
    );
  }

  return null;
},
);
}
}
```

Trong đoạn mã nguồn trên, phương thức *findById()* định nghĩa trong lớp **ProductsManager** (*lib/ui/products/products_manager.dart*) như sau:

```
class ProductsManager {
  ...
  Product? findById(String id) {
    try {
      return _items.firstWhere((item) => item.id == id);
    } catch (error) {
      return null;
    }
  }
}
```

- Định nghĩa widget **AppDrawer** thực hiện điều hướng trong ứng dụng (*lib/ui/shared/app_drawer.dart*):

```

1  import 'package:flutter/material.dart';
2
3  import '../orders/orders_screen.dart';
4  import '../products/user_products_screen.dart';
5
6  class AppDrawer extends StatelessWidget {
7    const AppDrawer({super.key});
8
9    @override
10   Widget build(BuildContext context) {
11     return Drawer(
12       child: Column(
13         children: <Widget>[
14           AppBar(
15             title: const Text('Hello Friend!'),
16             automaticallyImplyLeading: false,
17           ),
18           const Divider(),
19           ListTile(
20             leading: const Icon(Icons.shop),
21             title: const Text('Shop'),
22             onTap: () {
23               Navigator.of(context).pushReplacementNamed('/');
24             },
25           ),
26           const Divider(),
27           ListTile(
28             leading: const Icon(Icons.payment),
29             title: const Text('Orders'),
30             onTap: () {
31               Navigator.of(context)
32                 .pushReplacementNamed(OrdersScreen.routeName);
33             },
34           ),
35           const Divider(),
36           ListTile(
37             leading: const Icon(Icons.edit),
38             title: const Text('Manage Products'),
39             onTap: () {
40               Navigator.of(context)
41                 .pushReplacementNamed(UserProductsScreen.routeName);
42             },
43           ),
44         ],
45       ),
46     );
47   }
48 }
49

```

- Thêm drawer vào các trang, điều chỉnh điều hướng giữa các trang:
 - `lib/ui/products/product_grid_tile.dart`:

```

class ProductGridTile extends StatelessWidget {
  ...

  @override

```

```

Widget build(BuildContext context) {
  return ClipRRect(
    borderRadius: BorderRadius.circular(10),
    child: GridTile(
      ...
      child: GestureDetector(
        onTap: () {
          // Chuyển đến trang ProductDetailScreen
          Navigator.of(context).pushNamed(
            ProductDetailScreen.routeName,
            arguments: product.id,
          );
        },
        ...
      ),
    ),
  );
}

```

◦ *lib/ui/products/products_overview_screen.dart:*

```

...
import '../shared/app_drawer.dart';
...

class _ProductsOverviewScreenState extends State<ProductsOverviewScreen> {
  var _showOnlyFavorites = false;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('MyShop'),
        actions: <Widget>[
          ...
          ShoppingCartButton(
            onPressed: () {
              // Chuyển đến trang CartScreen
              Navigator.of(context).pushNamed(CartScreen.routeName);
            },
          ),
        ],
      ),
      // Thêm Drawer
      drawer: const AppDrawer(),
      body: ProductsGrid(_showOnlyFavorites),
    );
  }
}

```

◦ *lib/ui/products/user_products_screen.dart:*

```

...

```



```
import '../shared/app_drawer.dart';
...

class UserProductsScreen extends StatelessWidget {
  static const routeName = '/user-products';

  const UserProductsScreen({super.key});

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      ...
      // Thêm Drawer
      drawer: const AppDrawer(),
      body: const UserProductList(),
    );
  }
}
...
```

o *lib/ui/orders/orders_screen.dart*:

```
...
import '../shared/app_drawer.dart';
...

class OrdersScreen extends StatelessWidget {
  static const routeName = '/orders';

  const OrdersScreen({super.key});

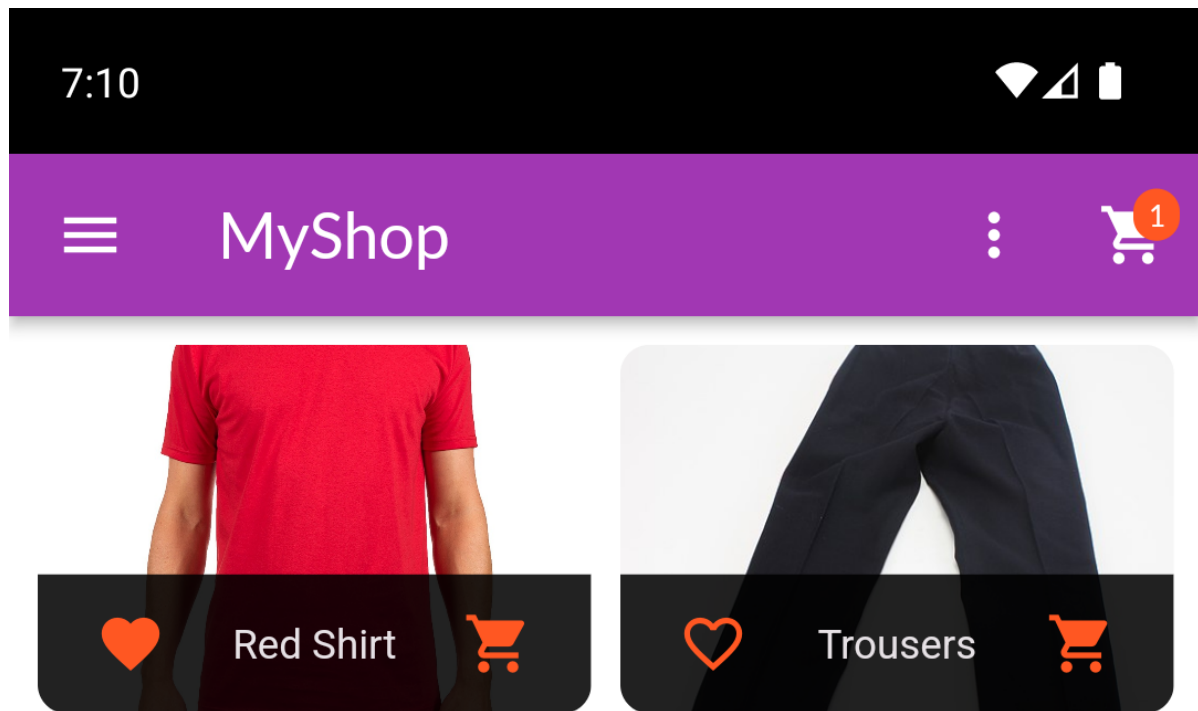
  @override
  Widget build(BuildContext context) {
    final ordersManager = OrdersManager();

    return Scaffold(
      appBar: AppBar(
        title: const Text('Your Orders'),
      ),
      // Thêm Drawer
      drawer: const AppDrawer(),
      ...
    );
  }
}
```

- Sau khi kiểm tra, lưu mã nguồn vào repo git và lên GitHub:

```
git add -u
git add lib/ui
git commit -m "Xây dựng drawer và danh sách theo tên"
git push origin master
```

Bước 4: Thêm thông tin số sản phẩm trong giỏ hàng



- Định nghĩa widget **TopRightBadge** (*lib/ui/products/top_right_badge.dart*) nhận vào một widget con và một giá trị. Giá trị sẽ được hiển thị ở góc trên bên phải của widget con:

```

1  import 'package:flutter/material.dart';
2
3  class TopRightBadge extends StatelessWidget {
4    const TopRightBadge({
5      super.key,
6      required this.child,
7      required this.data,
8      this.color,
9    });
10
11    final Widget child;
12    final Object data;
13    final Color? color;
14
15    @override
16    Widget build(BuildContext context) {
17      return Stack(
18        alignment: Alignment.center,
19        children: [
20          child,
21          Positioned(
22            right: 8,
23            top: 8,
24            child: Container(
25              padding: const EdgeInsets.all(2.0),
26              decoration: BoxDecoration(
27                borderRadius: BorderRadius.circular(10.0),
28                color: color ?? Theme.of(context).colorScheme.secondary,
29              ),
30              constraints: const BoxConstraints(
31                minWidth: 16,
32                minHeight: 16,
33              ),
34              child: Text(
35                data.toString(),
36                textAlign: TextAlign.center,
37                style: const TextStyle(
38                  fontSize: 10,
39                ),
40              ),
41            ),
42          ),
43        ],
44      );
45    }
46  }

```

- Hiệu chỉnh trang tổng quan các sản phẩm (*lib/ui/products/products_overview_screen.dart*) bao widget **IconButton** với widget **TopRightBadge**:

```

...
import '../cart/cart_manager.dart';
import 'top_right_badge.dart';
...

class ShoppingCartButton extends StatelessWidget {
  const ShoppingCartButton({super.key, this.onPressed});

```

```





























final void Function()? onPressed;

@override
Widget build(BuildContext context) {
  // Bao IconButton với TopRightBadge
  return TopRightBadge(
    data: CartManager().productCount,
    child: IconButton(
      icon: const Icon(
        Icons.shopping_cart,
      ),
      onPressed: onPressed,
    ),
  );
}
}

```

- Sau khi kiểm tra, lưu mã nguồn vào repo git và lên GitHub.

Cấu trúc thư mục dự án hiện tại:

- ▼  lib
 - ▼  models
 -  cart_item.dart
 -  order_item.dart
 -  product.dart
 - ▼  ui
 - ▼  cart
 -  cart_item_card.dart
 -  cart_manager.dart
 -  cart_screen.dart
 - ▼  orders
 -  order_item_card.dart
 -  orders_manager.dart
 -  orders_screen.dart
 - ▼  products
 -  product_detail_screen.dart
 -  product_grid_tile.dart
 -  products_grid.dart
 -  products_manager.dart
 -  products_overview_screen.dart
 -  top_right_badge.dart
 -  user_product_list_tile.dart
 -  user_products_screen.dart
 - ▼  shared
 -  app_drawer.dart
 -  dialog_utils.dart
 -  screens.dart
 -  main.dart